ПРАКТИЧЕСКАЯ РАБОТА 13

Тема: Файлы текстовые

студента 3ИСП-2

Эбулисова Эмиля

Ход работы:

Вариант 25

Задание 13-5: Следующие задания требуется решить с использованием текстовых файлов. Во всех заданиях предусмотреть в программе: формирование текстового файла, записав в него 20 случайных чиселот –10 до +10, по одному на строке. Используя созданный выше файл как входной, сформировать выходной файл по указанному для каждого варианта правилу. Записать выходной файл, разделив все нечетные по абсолютной величине числа на среднее арифметическое.

#include <fstream>

#include <random>

using namespace std;

int main() {

    // Создаем файл с 20 случайными числами от -10 до +10

    ofstream f\_out("first.txt");

    random\_device rd;

    mt19937 mt(rd());

    uniform\_int\_distribution<int> dist(-10, 10);

    int odd = -11; // Первое нечетное число

    for (int i = 0; i < 20; ++i) {

        int rnd = dist(mt);

        if (odd == -11 && rnd % 2 != 0)

            odd = abs(rnd);

        f\_out << rnd << '\n';

    }

    f\_out.close();

    // Используем созданный файл как входной

    ifstream f\_in("first.txt");

    ofstream s\_out("second.txt");

    while (f\_in.peek() != EOF) {

        int tmp;

        f\_in >> tmp;

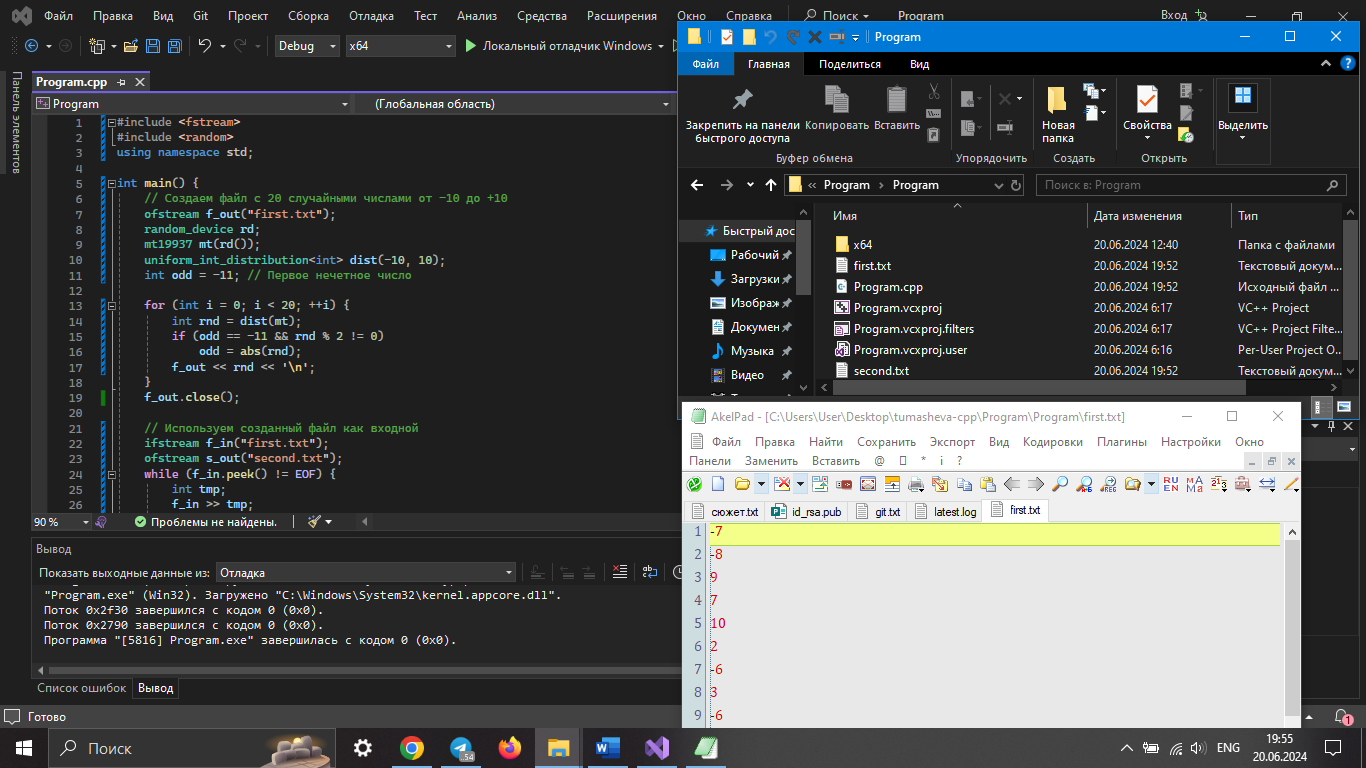
        s\_out << tmp + odd << '\n';

    }

    f\_in.close();

    s\_out.close();

}



Задание 13-11:Записать выходной файл, разделив все числа на половину максимального числа.

#include <fstream>

#include <random>

#include <algorithm> // для std::max\_element

using namespace std;

int main() {

// Создаем файл с 20 случайными числами от -10 до +10

ofstream f\_out("first.txt");

random\_device rd;

mt19937 mt(rd());

uniform\_int\_distribution<int> dist(-10, 10);

int odd = -11; // Первое нечетное число

for (int i = 0; i < 20; ++i) {

int rnd = dist(mt);

if (odd == -11 && rnd % 2 != 0)

odd = abs(rnd);

f\_out << rnd << '\n';

}

f\_out.close();

// Используем созданный файл как входной

ifstream f\_in("first.txt");

ofstream s\_out("second.txt");

int max\_num = \*max\_element(istream\_iterator<int>(f\_in), istream\_iterator<int>());

// Перемещаем указатель чтения обратно в начало файла

f\_in.clear();

f\_in.seekg(0);

while (f\_in.peek() != EOF) {

int tmp;

f\_in >> tmp;

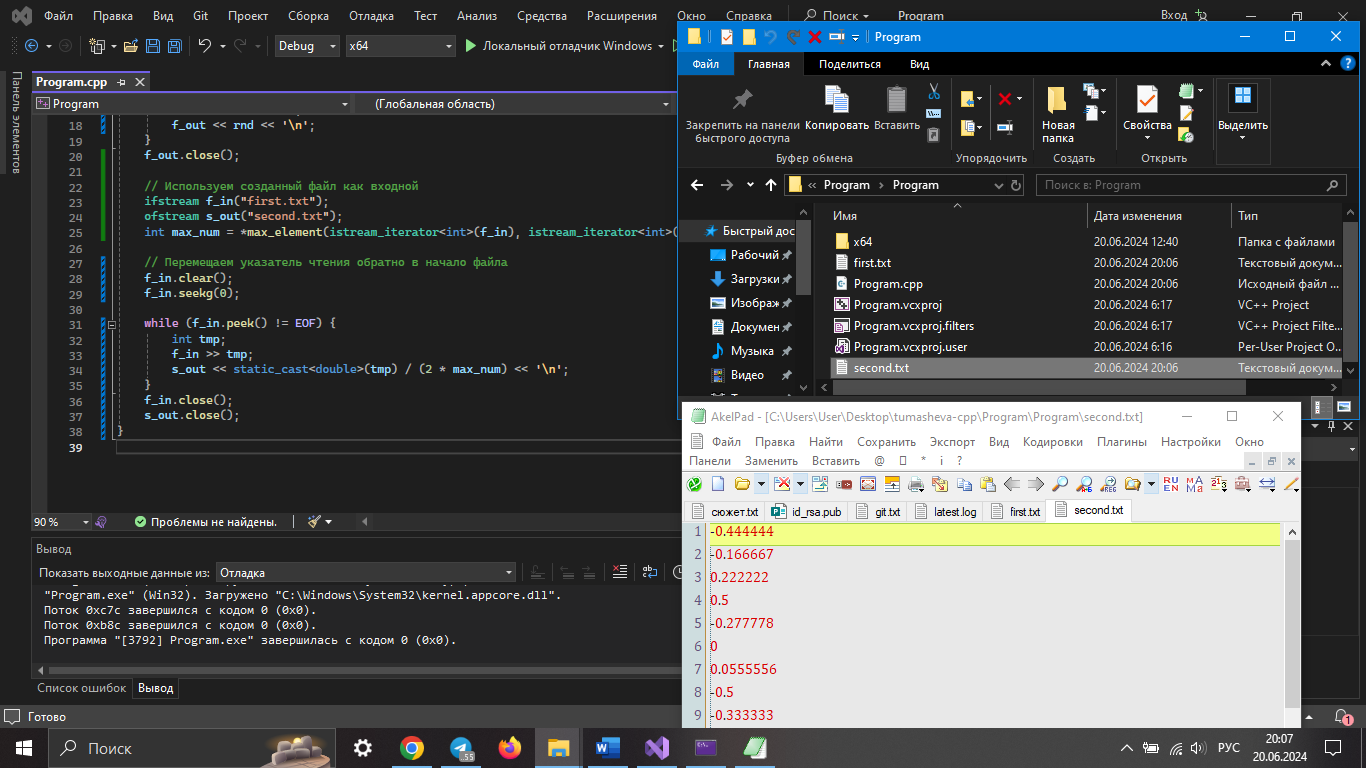
s\_out << static\_cast<double>(tmp) / (2 \* max\_num) << '\n';

}

f\_in.close();

s\_out.close();

}

Задание 13-25: Записать выходной файл, заменив каждое положительное число этим же числом, разделенным на максимальное из всех чисел.

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

int main() {

setlocale(LC\_ALL, "Russian");

// Открываем входной файл

std::ifstream input\_file("first.txt");

if (!input\_file.is\_open()) {

std::cerr << "Ошибка открытия файла!" << std::endl;

return 1;

}

// Читаем числа из файла в вектор

std::vector<double> numbers;

double num;

while (input\_file >> num) {

numbers.push\_back(num);

}

input\_file.close();

// Находим максимальное число

double max\_num = \*std::max\_element(numbers.begin(), numbers.end());

// Заменяем каждое положительное число

for (double& n : numbers) {

if (n > 0) {

n = n / max\_num;

}

}

// Открываем выходной файл

std::ofstream output\_file("second.txt");

if (!output\_file.is\_open()) {

std::cerr << "Ошибка открытия файла!" << std::endl;

return 1;

}

// Записываем обновленные числа в выходной файл

for (const double& n : numbers) {

output\_file << n << '\n';

}

output\_file.close();

std::cout << "Выходной файл успешно создан." << std::endl;

return 0;

}

